REMARKS

Claims 1-48 are pending in the application.

Claims 10-17 and 28-48 are canceled without prejudice or disclaimer in response to a restriction requirement.

Claims 1-9 and 18-27 are rejected.

The Applicants respectfully assert that the amendments to Claims 1-9 and 18-27 and incorporated by reference in any claims depending therefrom, are not narrowing amendments made for a reason related to the statutory requirements for a patent that will give rise to prosecution history estoppel. *See Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 122 S. Ct. 1831, 1839-40, 62 U.S.P.Q.2d 1705, 1711-12 (2002); 234 F.3d 555, 566, 56 U.S.P.Q.2d 1865, 1870 (Fed. Cir. 2001).

I. DRAWINGS

The Examiner objected to the drawings as failing to comply with 37 C.F.R. §1.85(p)(4) because in FIG. 2B the reference characters "220", "222", "223", and "224" have been used to designate two elements as shown in FIGS. 2F-2G. The Applicants have submitted a new drawing FIG. 2B replacing "220", "222", "223", "224" and "225" with "230", "232", "233", "234" and "235", respectively. The Specification is amended to conform to the new drawing FIG. 2B.

The Examiner objected to FIG. 3A as in reference number 309, "CUR_LSptr" should read -CUR_LS_ptr-. The Examiner objected to FIG. 3B as in reference numbers 330, 331, 333, 337, 334, and 335, "CUR_LSptr" should read -CUR_LS_ptr- and in reference numbers 329 and 336 "LStag" should read –LS_tag-. The Examiner objected to FIG. 4B as in reference numbers 426, 428, 431, 433, 434, 435, and 436, "LSptr" should read -LS_ptr- and in reference number 429 "OTQptr" should read –OTQ_ptr-. Similar

problems existed in FIGS. 4C-4G. The Applicants have corrected these types of informalities in the drawings.

The Applicants have submitted new drawings for FIGS. 2B, 2D-2E, 3A-3B, 4B-4E to correct additional informalities as suggested by the Examiner.

II. CLAIM OBJECTIONS

The Examiner objected to Claims 3-5 and 20-22 for the following formalities. In Claims 3 and 20, line 2, "an" should read –and-; and in Claims 4 and 21, it is unclear what is meant by "for a first type operation." The Examiner suggested changing "for a first type operation" to –for the operation of said first type instruction-. As all the claims are amended so the claim language conforms to the language in the Specification, these corrections were included.

III. REJECTION UNDER 35 U.S.C. § 112

The Examiner rejected Claims 1-9 and 18-27 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter.

The Examiner states that, in Claim 1, it is not clear what the real claimed invention is since the step for the first type instruction (PUSH, 246): "setting a first data value corresponding to a first address stored in said link stack in a fist portion of an entry in a queue having a plurality of entries" is also required of the second type instruction (POP, 249). Likewise, the Examiner states that "setting said current value of said first pointer in a second portion of said entry in said queue" is also required for the PUSH.

The invention of Claim 1 is detailed relative to FIGS. 3A-3C and the detailed description in the Specification. Claim 1 has been amended to improve the clarity of the language and to change "instruction" to "operation." In the Specification, "branch-to-link" (BrL) and "branch-to-link register (BrLR) are defined as instructions and adding addresses to a link stack (PUSH) and reading addresses from the link stack (POP) are

called "operations." Claim 1 is directed to operations used in managing a link stack. Further, Claim 1 has been amended adding that the first data value corresponds to "a portion of" a first address stored in the link stack. This clarifies what is meant by the phrase "first data value corresponding to."

Amended Claim 1 has threes steps. In step 1, a first data value corresponding to a first address stored in said link stack, into a first portion of an entry in a queue having a plurality of entries in response to a first type operation. The first type operation is a PUSH operation illustrated by actions taken out of step 325 in FIG. 3B. The first data value is the LS_tag from step 326. The LS_tag corresponds the middle portion of the PUSH address of the PUSH operation. This address is stored in the link stack at an address pointed to by the LS_ptr (e.g., 304) stored with the PUSH operation (defined in OPERATION_info 303) in the OTQ (e.g., OTQ 306). The first data value (e.g., LS_tag) is set into a first portion (e.g., LS_tag field 305) of an entry (register pointed to by ALLOCATE_ptr 302) in a queue (e.g., OTQ 306) having a plurality of entries (active operations 317).

In step 2 of Claim 1, a current value (address) of a first pointer (e.g., CUR_LS_ptr 309) of the link stack (e.g., LS 307) is set into a first register (e.g., register 310) in response to a second type operation. The second type operation is a POP operation illustrated by actions taken out of step 338 in FIG. 3B. In step 335 of FIG. 3B, the CUR LS_ptr is placed in the LS_CORRECTION_ptr (which is stored in register 310, See FIG. 3A).

In step 3 of Claim 1, the current value of the first pointer is set into a second portion of the entry of the queue setting said current value (address) of said first pointer (CUR_LS_ptr 309) in a second portion (LS_ptr field 305) of said entry (register pointed to by ALLOCATE_ptr 302) in said queue (e.g., OTQ 306). Step 3 is illustrated in step 333 of FIG. 3B and is executed for both the first and second type operation.

The above explanation along with the language amendments to Claim 1 point out that the invention of Claim 1 is not indefinite and has support in the Detailed Description page 17, line 20 through page 20, line 19, and FIGS. 3A-3F. Likewise, details of

elements of FIG. 3A that are similar to elements in FIG. 2A are referenced in these sections and explained in the Specification page 10, line 4 through page 17, line 19. Therefore, the Applicants respectfully assert that the rejection of Claim 1 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed by the above arguments.

Claim 18 is directed to a data processing system implementing the method steps of Claim 1. The Examiner rejected Claim 18 for the same arguments as Claim 1. Therefore, the Applicants respectfully assert that the rejection of Claim 18 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed for the same reasons as Claim 1.

Claim 2 is dependent from Claim 1 and contains all the limitations of Claim 1. Claim 2 adds two steps to Claim 1. The Examiner rejected Claims 2 and 19, line 4 because the Examiner states that "it is not clear which one is referred to as a second register." Claim 2 has been amended to change "instruction" to "operation" to be consistent with Claim 1 from which it depends. It should be clear from these amendments that Claim 2 is adding steps corresponding to the second type operation or POP operation. Again, actions of the POP operation are shown in the "YES" path out of step 338 in FIG. 3B. The first step of Claim 2 is detailed in this path at step 334. In step 334, the POP address (second address) is read from LS[CUR_LS_ptr]. LS[CUR_LS_ptr] is the register of LS 307 pointed to by the address in the current link stack pointer (LS_ptr). Therefore, the second address (POP address) stored at the stack entry (register of LS) defined by the current value of the first pointer (CUR_LS_ptr) is read. The second step of Claim 2 is detailed in this path at step 341. In step 2 of Claim 2, the second address is stored in a second register. The second address of Claim 2 is the POP address read in step 1 of Claim 2. In step 341 of FIG. 3B, the POP address is placed in the (LS_CORRECTION_add (link stack correction address). In FIG. 3A, LS_CORRECTION_add is stored in register 311; therefore, the second register is register 311 as shown in FIG. 3A.

The above explanation along with the language amendments to Claim 2 point out that the invention of Claim 2 is not indefinite and has support in the Detailed Description page 17, line 20 through page 20, line 19, and FIGS. 3A-3F. Likewise, details of elements of FIG. 3A that are similar to elements in FIG. 2A are referenced in these sections and explained in the Specification page 10, line 4 through page 17, line 19. Therefore, the Applicants respectfully assert that the rejection of Claim 2 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed by the above arguments.

Claim 19 adds the same limitations to Claim 18 that Claim 2 adds to Claim 1. The Examiner rejected Claim 19 for the same arguments as Claim 2. Therefore, the Applicants respectfully assert that the rejection of Claim 19 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed for the same reasons as Claims 1 and 2.

Claim 5 depends from Claim 4 and contains all the limitations of Claim 4. Claim 4 has been amended to correctly depend from Claim 2 where the second register is introduced. The Examiner rejected Claims 5 and 22 stating that "said second register in a said first portion of said entry" lacks antecedent basis since it was not defined previously. This antecedent basis problem is corrected by the amendment of Claim 4. Therefore, the Applicants respectfully assert that the rejection of Claim 5 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed by the above arguments and for the same reasons as Claim 2.

Claim 22 depends from Claim 21 and adds the same limitations to Claim 21 that Claim 5 adds to Claim 4. Therefore, the Applicants respectfully assert that the rejection of Claim 22 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed for the same reasons as Claims 5 and 2.

Claim 8 depends from Claim 7 which depends from Claim 2. Claim 8 has all the limitations of Claim 7 and Claim 2. The Examiner rejected Claim 8 because the Examiner states that "it is not clear how to compare said value from said retrieving step with a value form said second portion of said entry pointed to said pointer value since said value from said retrieving step is an address (said second address, see Claim 2, line 4) and a value from said second portion of said entry is a pointer value. Comparing an address with a pointer value was not described in the specification." Claim 8 has to read relative to FIGS. 3A-3F as these figures and their corresponding detailed description relate to the invention of Claim 1 and the ensuing dependent claims. Claim 8 depends from Claim 7. Claim 7 adds two steps to Claim 2. Claim 7 adds step 1 wherein a pointer value is received, in response to a pipeline flush, wherein the pointer value operable for pointing into the queue. It has been shown that the queue is the OTQ (e.g., OTQ 306). A pointer is defined in TechWeb on the internet as "in programming, a variable that holds the address of another variable or the address of the beginning of an array of variables. When a pointer to an array is used, it is made to point to any element in the array by incrementing or decrementing its contents (incrementing or decrementing the address)." A pointer is sometimes used interchangeable with the element (e.g., register or memory location) that holds an address to the other variable and the address itself (value of the pointer.) Therefore, a pointer (e.g., register) holds a particular (what is the pointer used for) pointer value (address) defining where a variable is stored. Therefore, for example, a link stack pointer (e.g., LS_ptr 309) contains the address (of a particular register) in the link stack where some variable (e.g., tracked addresses) are stored. As the pointer value (address in a pointer) is modified (e.g., incremented or decremented) the LS_ptr (its pointer value) points to a different element (e.g., register in the link stack) to which it is associated.

Step 1 of Claim 7 recites: receiving a pointer value (address) in response to a pipeline flush, wherein the pointer value is operable to pointing into the queue (OTQ 306). The mechanisms that are used to address OTQ 306 are the ALLOCATE_ptr 302, the DEALLOCATE_ptr 301, and the FLUSH_ptr 312. Therefore, the pointer value in

Claim 7 must be an address for one placed into one of these three pointers (registers for addressing the OTQ 306).

In step 2 of Claim 7, a current pointer value of the first pointer (CUR_LS_ptr) is set to a value in said second portion (LS_ptr field 304) of an entry (register in the OTQ 306) pointed to by said pointer value (address retrieved in step 1 of Claim 7). Since Claim 7 is directed to a flush operation, the details of the actions of Claim 7 are illustrated in FIG. 3C.

Therefore, Claim 7 recites, in response to a pipeline flush (OTQ FLUSH 389), receiving a pointer value operable for pointing into OTQ 306 (FLUSH_ptr 312). See step 391 in FIG. 3C. In step 382, the CUR_LS_ptr 309 (current pointer value of the first pointer) is set to a value in the second portion (value found in the LS_ptr field 304) of an entry pointed to by the pointer value (OTQ[FLUSH_ptr], entry of OTQ 306 pointed to by FLUSH_ptr 312).

Claim 8 adds two steps to Claim 7. Therefore, FIG. 3C is again referred to as illustrating the steps of Claim 8. In step 1 of Claim 8, a value is retrieved from the second register (register 311 holding the LS_CORRECTION_adr). In step 388, the value retrieved from register 311 is the LS_tag from the LS_CORRECTION_adr stored there. Therefore, the value that is retrieved from the second register is the LS_tag extracted from the LS_CORRECTION_adr. In step 2 of Claim 8, the value from said retrieving step (LS_tag from the LS_CORRECTION_adr) is compared with a value from the second portion (LS_tag field 305) of said entry (in OTQ 306) pointed to by said pointer value (FLUSH_ptr). Step 2 of Claim 8 is illustrated by step 383 in FIG. 3C

The above explanation of Claim 8 points out that the invention of Claim 8 is not indefinite and has support in the Detailed Description page 17, line 20 through page 20, line 19 and FIGS. 3A-3F. Likewise, details of elements of FIG. 3A that are similar to elements in FIG. 2A are referenced in these sections and explained in the Specification page 10, line 4 through page 17, line 19. Therefore, the Applicants respectfully assert

that the rejection of Claim 8 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed by the above arguments.

Claim 25 is directed to a data processing system implementing the method steps of Claim 8. The Examiner rejected Claim 25 for the same arguments as Claim 8. Therefore, the Applicants respectfully assert that the rejection of Claim 25 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter is traversed for the same reasons as Claims 2 and 8.

IV. REJECTION UNDER 35 U.S.C. § 102

The Examiner rejected Claims 1-9 and 18-27 under *35 U.S.C. §102(e)* as being anticipated by U.S. Patent No. 6,633,974 to *Sinharoy* (hereafter "*Sinharoy*").

For a reference to anticipate a claimed invention, the reference must disclose every aspect of the claimed invention. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). The identical invention must be shown in as complete detail as is contained in the claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989).

Claim 1 has been amended to more clearly define the present invention. The Examiner states that *Sinharoy* discloses Claim 1 and states for a first type operation (PUSH operation) that the first data value of Claim 1 is "Current Stack Operation 74, Fig. 3C of *Sinharoy*." Current Stack Operation 74 is in entry 57 of Branch Instruction Queue (BIQ) 56. See *Sinharoy*, FIG. 3A2. A stack operation is either a read (POP) or a store (PUSH) into a register stack. Therefore, Current Stack Operation 74, is a value defining whether the current stack operation called for in an entry 57 of BIQ 56 is either a PUSH or a POP. Since step 1 of Claim 1 is relative to only a first type operation, it cannot be either a PUSH or a POP. The Examiner states that the first type operation is a PUSH type operation. The Examiner states that the Current Stack Operation 74 corresponds to a Return Address 68 shown in FIG. 3D. *Sinharoy* states that "if the active stack has

changed for the write operation, then a data value ("change_bit"), which may be a one-bit indicator, is stored in the stack entry along with the return address. (This information indicates from where to Pop for the next entry once this current entry has been popped)." Therefore, according to *Sinharoy*, Return Address 68 is only associated with a POP type operation which is not the first type operation recited in Claim 1 of the present invention. Further, the Return Address 68 indicates where to POP (from an active link stack). See *Sinharoy*, column 5, lines 18-23. If the first data value is the Current Stack Operation 74 located in BIQ 56, it cannot be associated with Return Address 68 which indicates where to POP from link stack 58. Further, Claim 1 has been amended adding that the first data value corresponds to "a portion of" a first address stored in the link stack. This clarifies what is meant by the phrase "first data value corresponding to." This further points out that Return Address 68 of *Sinharoy* is not the first address stored in the link stack as recited by Claim 1.

The Examiner further states that the first portion of the entry is also Current Stack Operation 74 in BIQ 56. The first value of Claim 1 is set (stored) into the first portion of the entry in the queue. The Current Stack Operation 74 is part of a Branch instruction stored in BIQ 56. Branch instructions are decoded to determine what operations to perform. Nowhere does *Sinharoy* state that fields of branch instructions in BIQ 56 are written into response to an address stored in a link stack. Therefore, the Applicants assert that step 1 of Claim 1 is not disclosed by *Sinharoy* as stated by the Examiner.

Amended Claim 1 recites in step 2, setting a current value of a first pointer of said link stack into a first register in response to a second type operation. The Examiner states that *Sinharoy* discloses step 2 of Claim 1 and states that the second type operation is a POP operation. Further, the Examiner states that the first pointer of the link stack is either pointer 61a or 61b which contain values ptr0 and ptr1, respectively. Pointer 61a and 61b are registers containing addresses (pointer value) ptr0 and ptr1. Therefore, the Examiner is asserting that, in response to a POP operation, the current value of a first pointer (ptr0) is set (stored) into a first register (register 61a). Since this would be a

trivial operation, take the value in a register and store it back to the same register, the recitation of *Sinharoy* cited by the Examiner does not disclose the limitation of step 2 of Claim 1.

Step 3 of Claim 1 recites setting the current value of the first pointer into a second portion of the entry in the queue. The entry in the queue is the same entry recited in step 1 of Claim 1. The current value of the first pointer is the same value that was set into the first register in step 2. The second portion of the entry in the queue is another field in the same entry corresponding to the first portion recited in step 1 of Claim 1.

The Applicants have shown that *Sinharoy* does not disclose all the steps of Claim 1. Therefore, the Applicants respectfully assert that the rejection of Claim 1 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed by the above arguments.

Claim 18 has been amended to conform to the claim language in Amended Claim 1. Amended Claim 18 is an independent claim directed to a data processing system implementing the method steps of Claim 1. Claim 18 recites a data processing system comprising a central processing unit (CPU), wherein the CPU includes a link stack and first logic operable; the first logic setting a first data value corresponding to a portion of a first address stored in said link stack into a first portion of an entry in a queue having a plurality of entries in response to a first type operation, setting a current value of a first pointer into said link stack in a first register in response to a second type operation, and setting said current value of said first pointer in a second portion of said entry in said queue. The Applicants have shown that *Sinharoy* does not disclose the method steps implemented by the first logic of Claim 18. Therefore, the Applicants respectfully assert that the rejection of Claim 18 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the same reasons as Claim 1.

Claim 2 has been amended to conform to the claim language in Claim 1 from which it depends. Claim 2 is dependent from Claim 1 and contains all the limitations of Claim 1. Claim 2 adds two steps to Claim 1. In step 1 of Claim 2, a second address, stored at a stack entry at the current value of the first pointer, is read from the link stack in response to said second type operation. The Examiner states that as he best

understands *Sinharoy* discloses step 1 of Claim 2. The Examiner states that the second type operation is a POP operation. The Examiner states that the link stack is link stack 58. The Examiner then states that *Sinharoy* discloses reading from link stack 58 a second address stored at a stack entry at the current value of the first pointer without citing where in *Sinharoy* this discloser is found with the steps recited in Claim 1. In step 2 of Claim 2, the second address read from the link stack entry defined by the current value of the first pointer is stored in a second register. The Examiner states that the second register is inherent in *Sinharoy* as register 66 in link stack 66 or the instruction register inside the CPU. While the fact that a register may be found in *Sinharoy* and the register could be used for storing the second address read from the link stack entry defined by the current value of the first pointer, this does not inherently teach the invention of Claim 2 which includes all the steps of Claim 1. For the second type operation of Claim 2 four steps occur; 1) the current value of the first pointer of the link stack is set into the first register, 2) a second address is read from the link stack at the location defined by the current value of the first pointer, 3) the second address is stored in the second register, and 4) the current value of the first pointer is set into a second portion of the entry in the queue. Citing isolated disclosure of *Sinharoy* and stating that *Sinharoy* has registers that "inherently" could be used to store addresses does not disclose the steps recited in Claims 1 and 2. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where all the steps of Claim 1 and 2 are disclosed in combination. Therefore, the Applicants respectfully assert that the rejection of Claim 2 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the same reasons above and for the same reasons as Claim 1.

Claim 19 has been amended to conform to the claim language in Amended Claim 2. Amended Claim 19 is dependent form Claim 18 and adds the same limitation to Claim 18 as Claim 2 adds to Claim 1 with the addition of the second logic. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 18. Claim 19 was rejected for the same reasons as Claim 2. Therefore, the Applicants respectfully

assert that the rejection of Claim 19 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the same reasons as Claims 2 and 18.

Claim 3 has been amended to conform to the claim language of Claim 1 from which it depends. Claim 3 contains all the limitations of Claim 1. Claim 3 adds the limitation that the first type operation is a "push" type operation and the second type operation is a "pop" type operation. The Examiner rejected Claim 3 citing that *Sinharoy* teaches a PUSH operation and a POP operation in his Table 1 with no further explanation. The Applicants have shown that *Sinharoy* does not disclose all the steps of Claim 1 including the Examiner's assertion that the first type operation is a PUSH operation and the second type operation is a POP operation. Therefore, the Applicants respectfully assert that the rejection of Claim 3 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for the same reasons as Claims 1.

Claim 20 has been amended to conform to the claim language in Amended Claim 3. Amended Claim 20 is dependent form Claim 19 and adds the same limitation to Claim 19 as Claim 3 adds to Claim 1. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 18. Claim 20 was rejected for the same reasons as Claim 3. Therefore, the Applicants respectfully assert that the rejection of Claim 20 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the same reasons as Claims 3 and 18.

Claim 4 has been amended to conform to the claim language of Claim 2 from which it now depends. The dependency of Claim 4 was amended to correct an antecedent problem with Claim 5. Claim 4 contains all the limitations of Claim 2. Claim 4 adds the step of-setting the first data value in a third register for said first type operation. The Examiner stated that the first data value is the Current Stack Operation 74. Current Stack Operation 74 is in the entry 57 in BIQ 56. The Examiner then states that the third register of Claim 4 is the same as the first portion of an entry in the queue. Therefore step 1 of Claim 1 would be the same as the step added in Claim 4. The Applicants have not stated that the first portion of the entry in the queue is the same as

the third register. The reason the Applicants use different names for these elements is that they are different elements. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the invention of Claim 4. The Applicants have shown that *Sinharoy* does not disclose all the steps of Claim 2. Therefore, the Applicants respectfully assert that the rejection of Claim 4 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for the same reasons as Claims 2.

Claim 21 has been amended to conform to the claim language in Amended Claim 19 from which it now depends. Amended Claim 21 is dependent form Claim 19 and adds the same limitation to Claim 19 as Claim 4 adds to Claim 2 with the addition of the third logic. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the third logic. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 19. Claim 21 was rejected for the same reasons as Claim 4. Therefore, the Applicants respectfully assert that the rejection of Claim 21 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 2 and 19.

Claim 5 has been amended to conform to the claim language in amended Claim 2. Claim 5 depends from Claim 4 and contains all the limitations of Claim 4. Claim 5 adds the step of setting a second data value from the second register into the first portion of the entry in the queue in response to a third type operation. The second register is introduced in Claim 2 and is the register where the second address is stored and the second address is read from the link stack at the location pointed to by the current link stack pointer. The Examiner states that *Sinharoy* discloses the third type operation as a FLUSH operation and cites *Sinharoy*, column 5, line 52-53. In this recitation, *Sinharoy* states; "during a FLUSH, all the pointers and the information on which one of the logical stacks is active is retrieved from the BIQ (56). Claim 5 recites, that in addition to all the steps in Claims 4, 2, and 1, the step of setting a second data value from the second register into the first portion of the entry in the queue in response to a third type operation (not a POP or a

PUSH). Therefore, what the Examiner asserts is the second register in Claims 1, 2, and 4 must be consistent. If the second register is register 68 in a link stack storing a Return Address, then in Claim 2, the second address is read from the link stack and stored back into the link stack. If the second register is an Instruction register inside the CPU, then the second address is read from the link stack and stored in some undisclosed Instruction register. The Examiner has failed to point out where *Sinharoy* discloses that, in response to a second type operation (POP according to the Examiner), an address is read from the link stack and stored back in the link stack or an address is read from the link stack and stored in an particular Instruction register. This is important because Claim 5 of the present invention recites the step of setting a second value from the second register into the first portion of the entry of the queue in response to a third type operation (not a PUSH or a POP). Since register 68 in the link stack is used only for POP operations then the second register cannot be register 68 as stated by the Examiner. Since the Examiner has stated that the second register could be any Instruction register, then how does Claim 5 determine which register has the second address so that the second data value may be obtained so that it can be set into the first portion of the entry of the queue as recited in step 1 of Claim 1. Relative to Claim 1, the Examiner stated that the Current Stack Operation (field) 74 is the first portion of the entry of the BIQ. In Claim 5, the Examiner states that register 75a is the first portion of BIQ 56. This is a contradiction wherein the Examiner is asserting that two entirely different elements disclose the same limitation of Claims 1 and 5.

The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for Claim 5 because the Examiner has failed to specifically show that *Sinharoy* has disclosed all the limitations of Claim 5. Therefore, the Applicants respectfully assert that the rejection of Claim 5 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 2 and 4.

Claim 22 has been amended to conform to the claim language in amended Claim 21 from which it now depends. Amended Claim 22 is dependent form Claim 21 and

adds the same limitation to Claim 21 as Claim 5 adds to Claim 4 with the addition of the fourth logic. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the fourth logic. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 21. Claim 22 was rejected for the same reasons as Claim 5. Therefore, the Applicants respectfully assert that the rejection of Claim 22 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 5 and 21.

Claim 6 has been amended to conform to the claim language of Claim 1 and to more clearly define the claimed invention. Amended Claim 6 depends from Claim 1 and contains all the limitations of Claim 1. Claim 6 adds the limitation that all the steps of Claim 1 are performed in response to a fetch of an instruction using a corresponding one of said first type operation and said second type operation. Claim 6 defines the source of the first and second type operations as fetched instructions. The Examiner states that *Sinharoy* discloses a PUSH operation and a POP operation. While this is true, the Applicants have shown that *Sinharoy* does not disclose the invention of Claim 1 and therefore does not disclose the invention of Claim 1 with the limitations of Claim 6. Therefore, the Applicants respectfully assert that the rejection of Claim 6 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claim 1.

Claim 23 has been amended to conform to the claim language of Claim 18 and to more clearly define the claimed invention. Amended Claim 23 depends from Claim 18 and contains all the limitations of Claim 18. Claim 23 adds the same limitation to Claim 18 that 6 adds to Claim 1.. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 18. Claim 23 was rejected for the same reasons as Claim 6. Therefore, the Applicants respectfully assert that the rejection of Claim 23 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the same reasons as Claims 1 and 18.

Claim 7 has been amended to conform to the claim language of Claim 2 and to more clearly define the claimed invention. Claim 7 is dependent from Claim 2 and contains all the limitations of Claim 2. Claim 7 adds two additional steps to the invention of Claim 2. In step 1 of Claim 7, a pointer value operable for pointing into the queue is received in response to a pipeline flush. The queue was introduced in step 1 of Claim 1. Therefore the pointer value in Claim 7 is an address defining an entry or register in the queue of Claim 1. The pointer to the queue of Claim 7 is received in response to a pipeline FLUSH operation. In step 2, a current pointer value of the first pointer is set to (made equal to) a value in the second portion of an entry in the queue pointed to by the (received) pointer value in step 1 of Claim 7. The second portion of entries in the queue contain values of the pointer of the link stack set there by the steps of Claim 1.

The Examiner states that *Sinharoy* discloses Claim 7 and states that having a pointer value operable for pointing into (sic) BIQ 56 is an inherent step. The Applicants assert that step 1 of Claim 7 is defining that the pointer value received in response to the pipeline FLUSH operation is the specific pointer that points to entry in the queue of Claim 1. The Applicants are not claiming the functionality of a general pointer. Further, the Examiner states that *Sinharoy* discloses step 2 of Claim 7 and cites that *Sinharoy* discloses that the pointers (ptr0 and ptr1) to one of two active link stacks (58a or 58b) are stored in a branch instruction in entry 57 of BIQ. These pointers are used for PUSH or POP operations and would not used for a FLUSH operation as recited in Claim 7. The Examiner is stating that the pointer value received in response to a pipeline FLUSH operation is a pointer to an entry in the BIQ 56. Nowhere does *Sinharoy* disclose a pointer value to the BIQ is received in response to a pipeline FLUSH operation. The present invention discloses that the OTQ 306 has a FLUSH_ptr 312 that points to entries in the OTQ 306. One of ordinary skill in the art would not assume that in response to a FLUSH operation one would receive a pointer to a BIQ and retrieve pointers to a link stack used for PUSH and POP operations on the link stack. Further, the Applicants have shown that *Sinharoy* does not disclose Claim 2 of the present invention. Therefore, the

Applicants respectfully assert that the rejection of Claim 7 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 1 and 2.

Claim 24 has been amended to conform to the claim language of Claim 19 and to more clearly define the claimed invention. Claim 24 is dependent from Claim 19 and contains all the limitations of Claim 19. Claim 24 adds the same limitations to Claim 19 that Claim 7 adds to Claim 2 with the addition of the fifth logic and sixth logic. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the fifth logic and sixth logic. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 19. Claim 24 was rejected for the same reasons as Claim 7. Therefore, the Applicants respectfully assert that the rejection of Claim 24 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 7 and 19.

Claim 8 has been amended to conform to the claim language of amended Claim 7 and to more clearly define the claimed invention. Claim 8 is dependent from Claim 7 and contains all the limitations of Claim 7. Claim 8 adds two additional steps to the invention of Claim 7. In step 1 of Claim 8, a data value is retrieved from the second register. This means that in response to the pipeline FLUSH operation of Claim 7, a data value is retrieved from the second register that contains the second address stored there after the POP operation in Claim 2. In step 2 of Claim 8, the data value retrieved in step 1 of Claim 7 is compared to a data value from the second portion of the entry in the queue pointed to by the pointer value (that points to entries of the queue) retrieved in response to the pipeline FLUSH operation of Claim 7. The Examiner states that *Sinharoy* discloses the invention of Claim 8 and again states that *Sinharoy* inherently teaches that the second register can be either register 68 or an Instruction register. The Applicants have traversed this argument concerning the second register in arguments for Claim 5. The Examiner then states that *Sinharoy* teaches comparing the data value retrieved in step 1 of Claim 8 with a value from the second portion of the entry in the

queue. The only argument presented is that the Examiner states that field 65 and field 67 of entry 57 of the BIQ are the second portion of the entry in the queue. Nowhere does the Examiner state that *Sinharoy* performs the comparing step of Claim 8. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out the comparing step of Claim 8. Further the Applicants have shown that *Sinharoy* does not teach step 1 of Claim 8. Therefore, the Applicants respectfully assert that the rejection of Claim 8 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 1, 2 and 7.

Claim 25 has been amended to conform to the claim language of Claim 24 and to more clearly define the claimed invention. Claim 25 is dependent from Claim 24 and contains all the limitations of Claim 24. Claim 24 adds the same limitations to Claim 24 that Claim 8 adds to Claim 7 with the addition of the seventh logic and the eighth logic. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the seventh logic and the eighth logic performing the comparing step. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 24. Claim 25 was rejected for the same reasons as Claim 8. Therefore, the Applicants respectfully assert that the rejection of Claim 24 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 8 and 24.

Claim 9 has been amended to conform to the claim language of amended Claim 8 and to more clearly define the claimed invention. Claim 9 is dependent from Claim 8 and contains all the limitations of Claim 8. Claim 9 adds two additional steps to the invention of Claim 8. In step 1 of Claim 9, the current value of the first pointer is compared to a data value in the first register in response to a match in the comparing step of Claim 8. In step 2 of Claim 9, the data value in the second register is set into the link stack at a location pointed to by the current value of the link stack pointer decremented by one. The Examiner states that the steps of Claim 9 are disclosed by *Sinharoy* and states (sic) "inherent step such as that shown in Table 1, after the POP step in action 8,

the counter 2 is decremented by one from 9 to 8. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the two comparing steps of Claim 9. The Applicants have shown that *Sinharoy* does not disclose Claim 8 of the present invention. The Examiner has offered not arguments except that a POP operation in *Sinharoy* is accompanied by decrementing a counter. Therefore, the Applicants respectfully assert that the rejection of Claim 9 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 8.

Claim 26 has been amended to conform to the claim language of Claim 25 and to more clearly define the claimed invention. Claim 26 is dependent from Claim 25 and contains all the limitations of Claim 25. Claim 26 adds the same limitations to Claim 25 that Claim 9 adds to Claim with the addition of the ninth logic and the tenth logic. The Applicants assert that the Examiner has failed to make a *prima facie* case of anticipation for failing to specifically point out where *Sinharoy* discloses the ninth logic and the tenth logic performing the comparing steps. The Applicants have shown that *Sinharoy* does not disclose the invention of Claim 25. Claim 26 was rejected for the same reasons as Claim 9. Therefore, the Applicants respectfully assert that the rejection of Claim 24 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 8 and 25.

Claim 27 has been amended to conform to the claim language of Claim 18 and to more clearly define the claimed invention. Claim 27 adds the limitation that the system of Claim 18 further comprises system memory coupled to the CPU and operable for storing a program of instructions including the first type operations and the second type operations. The Examiner states that *Sinharoy* teaches the system of Claim 18 with the addition of system memory for storing a program of instructions including the first type and second type operations simply by stating that *Sinharoy* discloses RAM 14 in Fig. 1 and PUSH and POP operations. The Applicants have shown that *Sinharoy* does not disclose the system of Claim 18. The Examiner has not offered new arguments relative to Claim 27 that would lead one of ordinary skill in the art to conclude that *Sinharoy*

teaches the system of Claim 18. Therefore, the Applicants respectfully assert that the rejection of Claim 27 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy* is traversed for the reasons above and for same reasons as Claims 18.

V. CONCLUSION

FIGS. 2B, 2D-2E, 3A-3B ,4B-4E have been amended to correct informalities as suggested by the Examiner. Some of the drawings had additional informalities and have been amended to more clearly conform to the Specification and to comply with the Examiner's suggestions.

The Specification has been amended as suggested by the Examiner and to conform to the Drawings.

Claims 1-9 and 18-27 have been amended to correct informalities and so the claim language conforms to the Specification .

The Applicants have traversed the rejections of Claims 1-9 and 18-27 under *35 U.S.C. §112*, second paragraph, as failing to particularly point out and distinctly claim the subject matter which the applicants regards as the invention by amendments to the claim language to conform to the language of the Specification and by pointing out that the Examiner was looking to the wrong figures as support for the claims.

The Applicants have traversed the rejections of Claims 1-9 and 18-27 under *35 U.S.C. §102(e)* as being anticipated by *Sinharoy*.

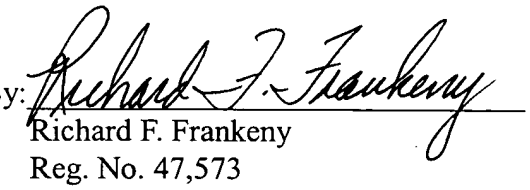
The Applicants, therefore, respectfully assert that Claims 1-9 and 18-27 are now in condition for allowance and request an early allowance of these claims.

Applicants respectfully request that the Examiner call Applicants' attorney at the below listed number if the Examiner believes that such a discussion would be helpful in resolving any remaining problems.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Patent Agent and Attorney for Applicants

By: 

Richard F. Frankeny
Reg. No. 47,573
Kelly K. Kordzik
Reg. No. 36,571

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2872

Austin_1\253858\1
7047-P293US 7/6/2004